УДК 681. 142

***Ю.И. ДОРОШЕНКО**, **А.В. ИВАШКО**, **А.В. ШОСТАК**,*
О.С. СОМХИЕВА (г. Харьков)

АЛГОРИТМ ГЕНЕРАЦИИ СТРУКТУР ДРЕВОВИДНЫХ ПСЕВДОМЕДИАННЫХ ФИЛЬТРОВ НА ОСНОВЕ ТЕОРИИ РАЗБИЕНИЙ

У статті наводиться алгоритм синтезу та програмна реалызацыя всых можливих структур *М*-арних дерев *D*(*M*, *N*) якы застосовуються для побудовы деревоподібного псевдомедіанного фільтру[3].

Paper is devoted a design of weighted tree pseudo median filter with minimal error of vector weghts and comparing with corresponding MAE.

Постановка проблемы. С развитием аппаратной части систем регистрации, обработки и визуализации изображений одновременно повышаются требования к характеристикам изображений: разрешению, глубине цвета, переходу к 3-*D* визуализации. Такие изображения требуют большой размер апертуры фильтров при обработке. Существующие вычислительные мощности персональных и специализированных компьютеров не позволяют проводить цифровую обработку, включающую фильтрацию, потокового видео и 3-*D* изображений.

Анализ литературы. Медианная фильтрация[1] является широко используемым методом нелинейной цифровой фильтрации для обработки изображений, который обладает важным достоинством – сохранением перепадов яркости изображения при наилучшем подавлении импульсных шумов. Поэтому, несмотря на большое количество методов поиска медианы, продолжаются поиски более эффективных[1].

В [2] авторами предложен древовидный псевдомедианный фильтр (ДПМФ), структура которого основана на М-арных деревьях[3]. Предложенный фильтр имеет более высокое быстродействие, чем стандартные медианные и взвешенные медианные фильтры, и одновременно имеет априорные вероятности прохождения входных отсчетов по позициям апертуры.

В данной статье рассматривается алгоритм генерации структур ДПМФ на основе раздела комбинаторики – теории разбиений[4].

М-арное дерево определим рекурсивно на основе элементарного *М*-арного дерева с одним узлом с *М* листами или входами. Если $A_1, A_2,..., A_M$ – *М*-арные деревья, то и $A = \{A_1, A_2,..., A_M\}$ является *М*-арным деревом.

Если элемент A_i является тривиальным деревом, то будем говорить о **листе** или **входе** дерева, если же элемент A_i не является тривиальным деревом, то будем говорить об **узле** дерева. **Порядком** *М*-дерева назовем количество его элементарных поддеревьев (базисных элементов). Очевидно, что порядок элементарного дерева равен единице: $|1| = 1$. Условимся обозначать *М*-дерево порядка *N* через *D(M, N)*. Если *А* – *М*-арное дерево, то его порядок условимся обозначать $|A|$.

Очевидно, что порядок дерева, состоящего из совокупности *М*-арных деревьев, равен сумме порядков этих деревьев, т.е. справедлива формула:

$$\left|\{A_1, A_2, \ldots, A_M\}\right| = \sum_{i=1}^{M} |A_i| \qquad (1)$$

Очевидно, что число узлов D(*M*, *N*) дерева можно подсчитать в соответствии со следующим выражением:

$$\Psi(M,N) = (N-1)/(M-1) \qquad (2)$$

Из формулы (2) следует, что, для существования хотя бы одного дерева *D*(*M*,*N*) необходимо и достаточно, чтобы (*M*−1) делило (*N*−1) нацело.

Рассмотрим задачу генерации всех неизоморфных *D*(*M*,*N*) деревьев.

Назовем ?–набором набор чисел $\{\alpha_i\}_{i=1}^{M}$, обладающих следующими свойствами:

1. Все значения $\{\alpha_i\}_{i=1}^{M}$ расположены в порядке невозрастания: $\alpha_1 \geq \alpha_2 \geq \ldots \geq \alpha_M$;
2. Сумма всех чисел набора равна *N*: $\alpha_1 + \alpha_2 + \ldots + \alpha_M = N$;
3. (M – 1) делит каждое из чисел набора: $M-1 | \alpha_i,\ i = \overline{1, M}$.

При этом число *М* будем называть длиной ?-набора. Два ?-набора длины *М* $\{\alpha_i\}_{i=1}^{M}$ и $\{\beta_i\}_{i=1}^{M}$ будем называть равными или одинаковыми**,** если у них совпадают соответствующие значения элементов, т.е. $\alpha_i = \beta_i, i = \overline{1, M}$. Тогда для любого *М*-арного дерева $A = \{A_1, A_2,..., A_M\}$ найдется изоморфное ему дерево $B = \{B_1, B_2,..., B_M\}$, такое, что значения $\{|B_1|, |B_2|,\ldots |B_M|\}$ образуют ?-набор. Для любого дерева указанный ?-набор определяется единственным образом. Существование и единственность ?-набора для указанного *М*-арного дерева дает нам возможность говорить о ?-наборе данного дерева.

Два *М*-арных дерева назовем изоморфными**,** если совпадают их ?-наборы. Если двум М-деревьям соответствуют разные ?-наборы, то они неизоморфны. Таким образом, это приводит нас к следующей идее построения всех неизоморфных *D*(*M*, *N*) деревьев:

1. Если *N* = 1, то единственным деревом будет являться тривиальное дерево.
2. Сгенерировать и рассмотреть все ?-наборы длины *М* и суммой *N*. Предположим, что $\{\alpha_i\}_{i=1}^{M}$ – текущий рассматриваемый набор.
3. Если все элементы $\alpha_i = 1, i = \overline{1, M}$, , то единственным *М*-деревом будет $\{1,1,\ldots\}$ длиной *М*.

4. Всем значениям α_i = 1 поставим в соответствие тривиальные деревья. Для остальных же значений (α_i > 1) построим все возможные *M*-деревья, пользуясь этим же алгоритмом.

5. Рассматривая все сочетания сгенерированных деревьев (при этом каждое из деревьев, соответствующее α_i должно занимать *i*-ю позицию), мы получим всякий раз *D*(*M, N*) дерево.

Указанный алгоритм гарантирует, что всякое *D*(*M, N*) дерево изоморфно хотя бы одному из сгенерированных. Однако он не гарантирует того, что все сгенерированные деревья будут неизоморфными. Например, в случае *M* = 3 и *N* = 15 одним из допустимых ?-наборов будет 7 + 7+1. Поскольку всего существует два различных неизоморфных дерева 7-го порядка (обозначим их 7a и 7b) то, в соответствии с алгоритмом, всего будет рассмотрено четыре варианта {7a, 7a, 1}, {7a, 7b,1}, {7b, 7a, 1}, {7b, 7b, 1}. Однако деревья {7a, 7b, 1} и {7b, 7a, 1} изоморфны между собой. Выходом состоит следующая модификация предложенного алгоритма таким образом, чтобы он при рассмотрении очередного α_i не генерировал деревья, которые были рассмотрены для предыдущих значений *i*.

Опишем алгоритм получения всех *?*-наборов заданной длины для *M*=3.

Пусть дано число *M,* и нам необходимо получить все *?*-наборы указанной длины. Предположим, что мы успешно сгенерировали значения α_1, α_2,..., α_k, и нам осталось получить значения α_{k+1}, …., α_M, $(0 \le k \le M)$. Если $k < M$, то необходимо установить α_{k+1} последовательно в α_k, $\alpha_{k-(M-1)}$, … , 1 (в случае *k* = 1 полагаем, что $\alpha_0 = N - (M - 1)$), и для каждого значения α_{k+1} продолжить поиск. Если *k* = *M,* то указанный набор сформирован и осталось только проверить его значение суммы.

Алгоритм получения μ–наборов *A*1 выполняет разложение целого нечетного числа *N* на *M* = 3 целых нечетных числа *A, B* и *C*, причем $A \ge B \ge C$ и $N=A+B+C$. Алгоритм *A*1 использует приводимый ниже алгоритм *A*2, который выполняет разложение четного целого числа *H* на два нечетных целых чисел *B* и *C*, причем $B \ge C$. В алгоритмах *A*1 и *A*2 число *p* определяет количество разложений числа *N*.

Входными данными для алгоритма *A*1 являются числа *N* и *M*=3. Выходными данными для алгоритма *A*1 являются числа *A*, *B*, *C* и *p*, причем число *A* определяется в алгоритме *A*1, числа *B* и *C* определяются в алгоритме *A*2, а число *p* формируется в алгоритмах *A*1 и *A*2.

Алгоритм *A*1 имеет следующий вид:

1. (Установка исходных значений.) $k = M - 1$, $p = 1$.

2. (Вычисление максимального числа $A = l$ при текущем *p*) $l = N - p \cdot k$.

3. (Разложение числа *N* на две части.) $N = c \cdot l + d$, где *c* – целая часть от деление *N* на *l*, *d*=*N* mod *l* – остаток от деления *N* на *l*.

4. (Анализ величин *c* и *d*) Если *c* = 1, то: - *A* = *l*; - передать в алгоритм *A*2 значения *A*, *H* = *d*, и *p*; - в результате работы алгоритма *A*2 получим наборы

B, C и *p*. Если *c* = 2 (при этом обязательно $d \neq 0$), то: - *A* = *l*; - передать в алгоритм *A*2 значения *A*, *H* = *l*+*d*, и *p*; - в результате работы алгоритма *A*2 получим наборы *B, C* и *p*. Если *c*= 3 и *d*=0, то: - *A* = *l*, *B* = *l*, *C* = *l*; - конец работы алгоритма *A*1. Если *c* = 3 и *d*>0, то: - разложение *N* в базисе *l* на *M* чисел отсутствует; - конец работы алгоритма *A*1. Если *c*>3 (больше *M*), то: - разложение *N* в базисе *l* на *M* чисел отсутствует; конец работы алгоритма *A*1.

5. *p* = *p*+1 и переход на пункт 2.

В алгоритме *A*2 *H* – четное целое число ($H \ge 2$), которое требуется разложить на два нечетных целых числа *B* и *C*, причем $B \ge C$ и $H = B+C$. Известно также нечетное целое число *A* ($A \ge 1$), для которого должно выполняться условие – $A \ge B \ge C$. То есть число *A* ограничивает количество разложений *p* числа *H* на два числа *B* и *C*.

Входными данным для алгоритма *A*2 являются числа *A*, *H*, и *p*. Выходными данными для алгоритма *A*2 являются числа *A*, *B*, *C* и *p*. Причем алгоритм *A*2 формирует только числа *B*, *C* и *p*.

Алгоритм *A*2 имеет следующий вид:

1. (Анализ входных данных.) Если *H* = 2, то: – *B* = 1, *C* = 1; – вывод *A*, *B*, *C* и *p*; – конец работы алгоритма *A*2.

2. (Анализ входных данных.) Если *H*>2, то: – *l* = *H*–1 (величина *l* определяет значение числа *B*); – *l* = min(*l*, *A*).

3. (Разложение числа *H* на две части.) $H = c \cdot l + d$, где – *c* = [*H*/*l*] – целая часть от деление *H* на *l*; – *d* – остаток от деления *H* на *l*.

4. (Анализ величин *c* и *d*.) Если *c* = 1, то: – *B* = *l*, *C* = *d*; – вывод *A*, *B*, *C* и *p*. Если *c* = 2 и *d* = 0, то: – *B* = *l*, *C* = *l*; – вывод *A*, *B*, *C* и *p*; – конец работы *A*2. Если c = 2 и d ≠ 0, то: – конец работы A2.

5. (Анализ величины *l*.) Если *l* = 3, то конец работы алгоритма *A*2. Иначе (при *l* >3): – *l* = *l*–2; – *p* = *p*+1; – переход на шаг 3.

Итак, алгоритм получения всех неизоморфных *D*(*M*,*N*) деревьев, как мы уже видели, несложен для понимания, однако его реализация требует определения способа представления *M*-деревьев в памяти. Один из способов заключается в том, чтобы для каждого узла дерева записывать число входов в него (всего *M* значений). При этом все листы будут иметь номера с 1 до *M* включительно, а все узлы будут нумероваться от *M* + 1 до *M* + (N − 1)/(*M* − 1). Таким образом, *D*(3, 5) может быть записано как (1, 2, 3), (6, 4, 5), а два неизоморфных дерева *D*(3, 7) как (1, 2, 3), (8, 4, 5), (9, 6, 7) и (1, 2, 3), (4, 5, 6), (8, 9, 7). Будем называть такое представление *D*(*M, N*) строкой. На основе метода генерации D(*M, N)* деревьев, изложенного выше на языке *Turbo Pascal* в среде *Delphi* была написана программа генерации структур ДПМФ.

Для генерации всех вариантов структур ДПМФ необходимо задать *M* (размер окна эквивалентного одномерного фильтра) и *N* (число входов базового элемента). Примеры работы программы приведены на рис. 1–4. На рис. 1–2 представлены копия экрана при генерации структур ДПМФ, получаемых из $D(11, 3)$. Все нечетные разбиения числа 11 на три части представлены в окне программы (рис. 2). Результаты для $D(25,5)$ изображены на рис. 3, 4.

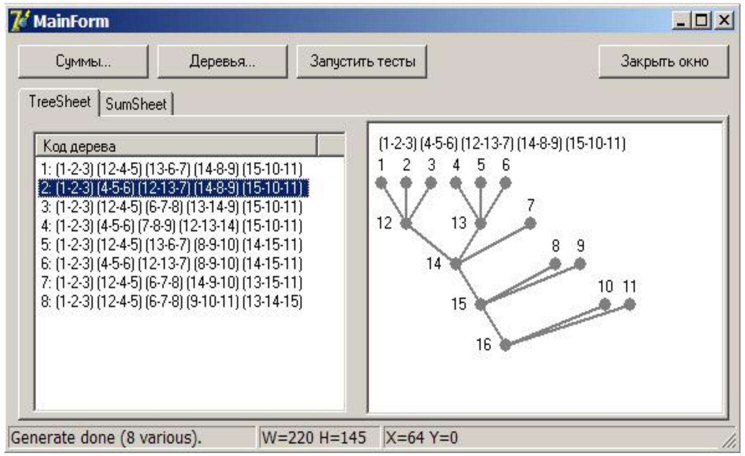

Рис.1. Структуры ДПМФ при $N = 11$, $M = 3$

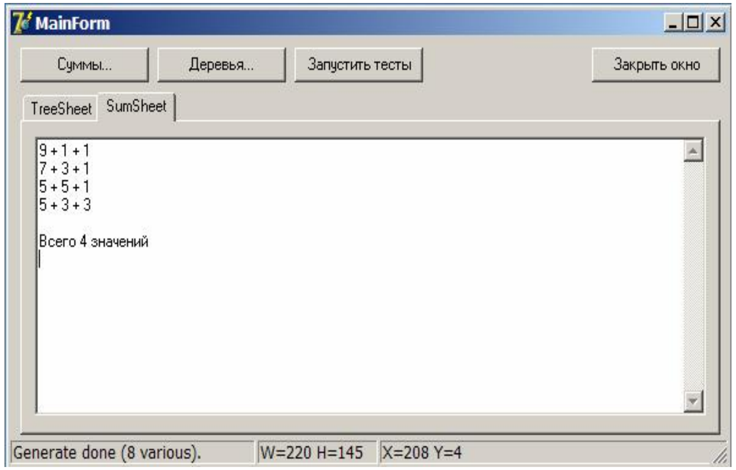

Рис.2. Разбиения числа 11 на три нечетные части

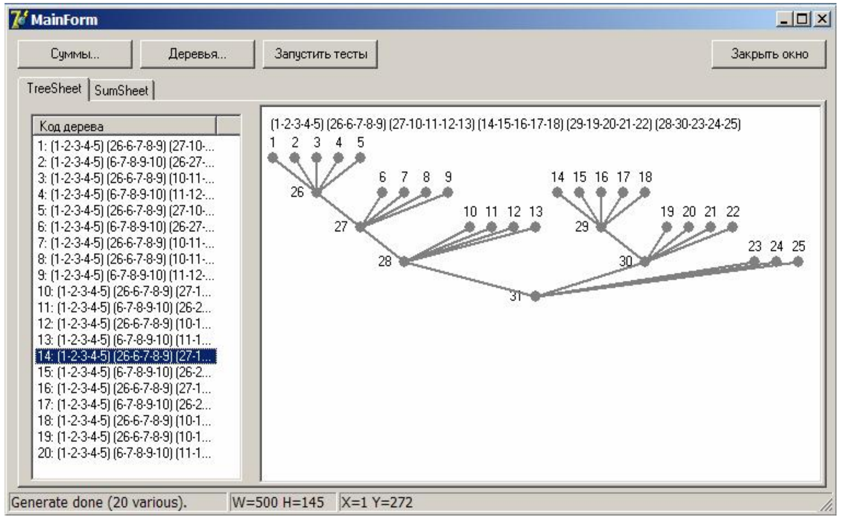

Рис. 3. Структуры ДПМФ при $N = 25$, $M = 5$

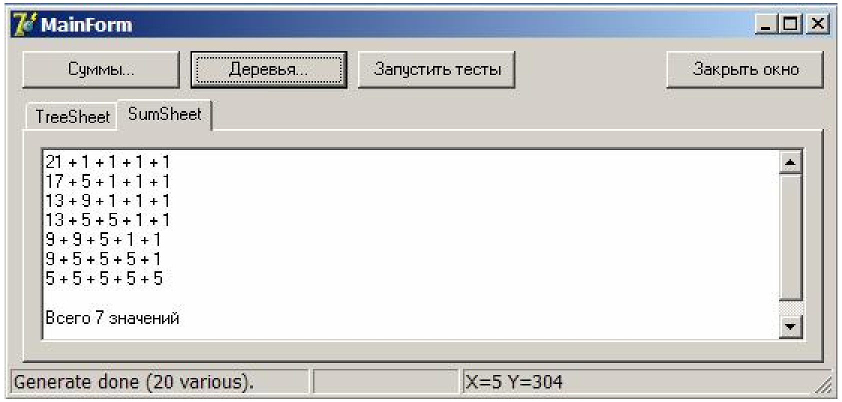

Рис. 4. Разбиения числа 25 на пять нечетных частей

Список литературы: 1. Быстрые алгоритмы в цифровой обработке изображений/ Под ред. Т. С. Хуанга, - М.: Радио и связь, 1984. - 224 с. **2.** *Дорошенко Ю.И.* Анализ и моделирование нелинейных цифровых фильтров на основе ранговых статистик // Вестник ХПУ, №21’97. Применение вычислительных систем. Вып.2. **3.** *Касьянов В.Н., Евстигнеев В.А.* Графы в программировании: обработка, визуализация и применение. - СПб.: БХВ-Петербург, 2003. - 1104 с.**4.** *Эндрюс Г.* "Теория разбиений".– М.: Наука, 1982. –256с.

Поступила в редколлегию 30.11.07